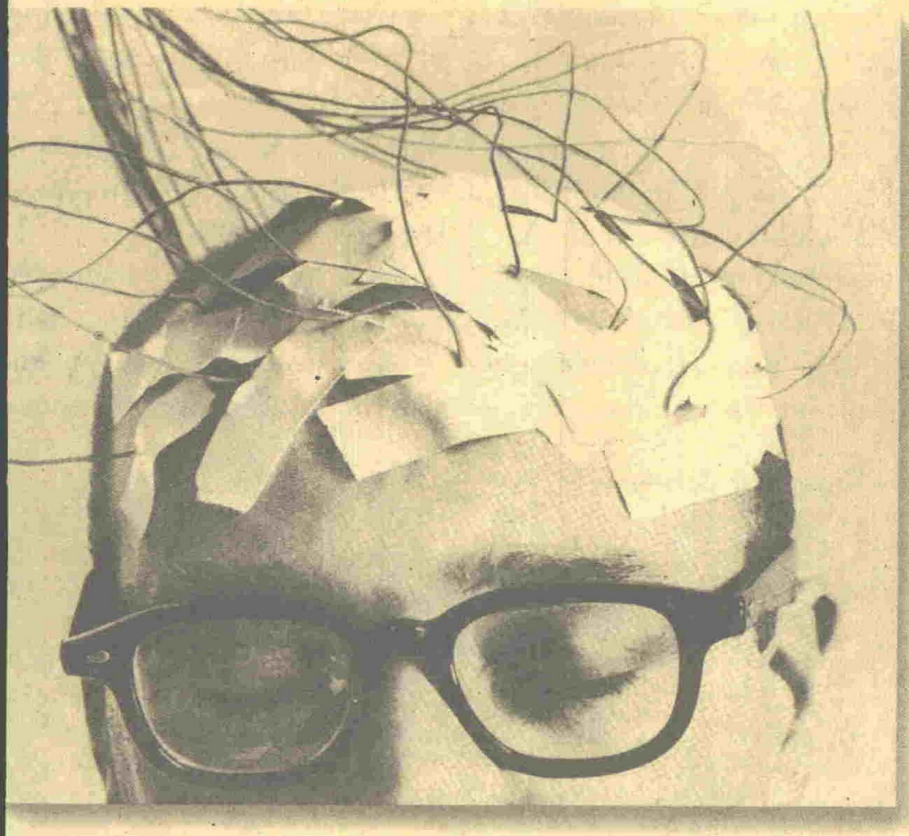
เทคนิคน่ารู้การโปรแกรมภาษาปาสคาล

การโปรแกรมแบบมองเป็นโพรซีเดอร์ของปาสคาลช่วยลดความยุ่งยากซับซ้อนในการเขียนโปรแกรม



ดร.ปัญญา เปรมปรีดิ์

ผมคิดว่าภาษาปาสคาลกำลังเริ่มเป็นที่นิยมของคนไทยเรา เรื่องนี้จะเห็นได้จากการมีตำราเรื่องภาษาเทอร์โบปาสคาลออกมาสู่ตลาดเมืองไทยเป็นจำนวน 4-5 เล่มในเวลาเพียง 2-3 เดือน นอกจากนี้บริษัทผู้ค้าเครื่องไมโครคอมพิวเตอร์ก็เปิดหลักสูตรสอนภาษานี้กันขึ้นเกือบทุกแห่ง ตามศูนย์การค้าต่าง ๆ เราจะเห็นโฆษณาเหล่านี้ เด็ก ๆ ที่นั่งอ่านตำราอยู่ตาม Foodmart ต่างก็พูดถึงภาษานี้กันอย่างกว้างขวาง ผมจึงเห็นว่าเทคนิคและข้อแนะนำเล็ก ๆ น้อย ๆ ในการใช้ภาษานี้คงจะมีประโยชน์อยู่บ้าง วันนี้ผมอยากจะพูดถึงเรื่องการเขียนโปรแกรมในแนว "ค่อยทำค่อยไป" ซึ่งเป็นวิธีการที่เหมาะกับภาษาปาสคาลเป็นอย่างยิ่ง

ความหมาย"ค่อยทำค่อยไป"

การเขียนโปรแกรมนั้น ถ้าเราสามารถจะเขียนแต่เฉพาะโครงร่างใหญ่เสียก่อน และทดสอบว่าโปรแกรมมันวิ่งไปตามส่วนต่าง ๆ ของโครงร่างนี้ได้มันก็จะดีมาก เพราะเราจะไม่ต้องพะวงกับรายละเอียด ตัวโครงร่างจะยาวไม่เท่าใดนัก เราจะสามารถทำให้ไม่มี Syntax Error ได้โดยเร็ว แล้วหลังจากนั้นเราจึงค่อยมาเติมรายละเอียดในส่วนต่าง ๆ ของโปรแกรม การเติมรายละเอียดนี้ก็เช่นกัน เราจะเติมทีละส่วน ทำแล้วทดสอบให้ดีเสียก่อน แล้วจึงจะไปเติมรายละเอียดในส่วนอื่น ถ้าเราทำอย่างนี้ได้ การพัฒนาโปรแกรมก็จะทำได้เร็ว โอกาสที่จะบั๊กหลงเหลืออยู่ก็จะมีน้อยมากและจะเพิ่มเติมหรือดัดแปลงในอนาคตก็ง่าย

นั่นแหละครับที่ผมเรียกว่า เขียนโปรแกรมแบบ "ค่อยทำค่อยไป" คำนี้ฟังดูเหมือนว่าจะพาให้ท่านคลานเป็นเต่า แต่จากประสบการณ์ของผมพบว่ามันไปได้สวยและไปได้เร็วกว่าเก่าอย่างมากมาย

ทำไมปาสคาลจึงเหมาะกับหลักการ "ค่อยทำค่อยไป"

ภาษาปาสคาลเป็น Highly Procedural Language หรือคือ เราจะเขียนคำสั่งอยู่ในตัวเมนบล็อกเพียงไม่กี่คำสั่ง มันจะเป็นตัวโครงร่างใหญ่ของโปรแกรม ในโครงร่างใหญ่อันนี้เราจะมีการเรียกใช้โพรซีเดอร์ (Procedure) จำนวนหนึ่ง แต่โพรซีเดอร์เหล่านี้เรามาเขียนรายละเอียดลงไปได้ในภายหลัง นอกจากนี้แล้วในโพรซีเดอร์เหล่านี้ถ้าเราเห็นว่ามันมีขนาดใหญ่เกินไป เรา

ก็อาจตัดแยกเป็นโพรซีเดอร์ย่อยได้อีก ในโพรซีเดอร์ใหญ่จึงไม่ต้องเขียนรายละเอียด เราเปลี่ยนมาเป็นการเรียกใช้โพรซีเดอร์ย่อย เมื่อเขียนและทดสอบโครงร่างของเมนบล็อกและโพรซีเดอร์ใหญ่เรียบร้อยแล้ว เราจึงค่อยไปทำรายละเอียดของโพรซีเดอร์ย่อย

เข้าถึงตัวอย่างกันเลยก็คงจะดี

โปรแกรมทางธุรกิจนั้นส่วนใหญ่จะเริ่มจากการขึ้นภาพบนจอให้เป็นเมนู คือให้ผู้ใช้เลือกว่าจะทำอะไร เมื่อผู้ใช้เลือกแล้วโปรแกรมก็จะไปเรียกเอาโพรซีเดอร์ที่ต้องการมาทำงาน เมื่อทำเสร็จแล้วก็จะกลับไปสู่เมนูอันเดิม มันจะวนอยู่อย่างนี้จนกว่าผู้ใช้จะเลือก "จบงาน" นี่คือโครงร่างใหญ่ของโปรแกรม ยกตัวอย่างเช่น เมนูอาจบอกให้เลือกได้ 6 อย่างดังนี้

(1) UPDATE MASTER FILE

(2) SCAN MASTER FILE

(3) MAKE REPORT (S)

(4) MAKE BACK-UP

(5) EXTRACT/RELOAD MASTER FILE

(6) END JOB

ซึ่งเราจะเห็นว่า ทางเลือกที่ 1 ถึง 5 นั้นเราสามารถจัดเป็นโพรซีเดอร์ใหญ่ที่แยกต่างหากออกจากกันได้เลย ส่วนทางเลือกสุดท้ายก็ไปสู่การจบโปรแกรม และเพื่อให้ง่ายต่อการเขียนโปรแกรม เราก็อาจตั้งชื่อโพรซีเดอร์ทั้ง 5 ไปตามลักษณะงาน คือ UPDATEMASTER, SCANMASTER, MAKEREPORT, MAKEBACKUP และ EXTRACTRELOAD ตามลำดับ จากนี้เราก็นำไปเขียนโครงร่างของโปรแกรมได้ รูปที่ 1 แสดงตัวอย่างของโปรแกรมที่เป็นโครงร่างดังกล่าวนี้

สิ่งที่ผมอยากให้ท่านสังเกตก็คือ ในโพรซีเดอร์ทั้ง 5 นั้นเราใส่คำสั่งที่จะบอกให้เราทราบว่าโปรแกรมได้ผ่านมาสู่โพรซีเดอร์ดังกล่าวจริง แต่ในชั้นต้นนี้เรายังไม่ทำอะไรเลย เราเพียงแต่ใช้คำสั่ง GOTOXY และ WRITE เพื่อบอกว่าโปรแกรมได้ผ่านมาสู่โพรซีเดอร์นี้ และใช้คำสั่ง REPEAT UNTIL KEYPRESSED เพื่อให้มันหยุดอยู่จนกว่าเราจะเคาะแป้นอันใดอันหนึ่ง และก่อนที่มันจะกลับไปสู่เมนเมนูเราก็ใช้คำสั่ง REPEAT READ (KBD, CH) UNTIL NOT KEYPRESSED เพื่อเคลียร์บัฟเฟอร์ของคีย์บอร์ดให้หมด

เห็นไหมครับ มันเล่นไม่ยาก เรา
สามารถทำโครงร่างของโปรแกรมเสร็จ
ในไม่กี่นาที ท่านสามารถคอมพายล์แล้ว
ทดสอบให้เป็นที่พอใจได้ โปรแกรมจะ
วนไปสู่ส่วนต่าง ๆ แล้วกลับมาที่เมนู
อันเดิมอย่างปลอดภัยทุกครั้ง หลังจาก
นั้นท่านก็สามารถเติมรายละเอียดไปที่
ละโพรซีเดอร์ได้ ใน 5 โพรซีเดอร์ที่มีอยู่
ผมแนะนำให้ทำ SCANMASTER
ก่อน ทั้งนี้เพื่อจะได้เอาไว้ใช้ตรวจผลการ
UPDATE ในตอนเขียนโพรซีเดอร์
UPDATEMASTER

แต่ก่อนจะเข้าถึงรายละเอียดของ
SCANMASTER ผมก็อยากจะพูดถึง
ลักษณะของมาสเตอร์ไฟล์เสียก่อน สม-
มติว่าเรื่องนี้เป็นเรื่องการเก็บรายละเอียด

รูปที่ 1 โครงร่างของโปรแกรม

```
PROGRAM TP1S01;
LABEL MAINLOOP,EOJ;
VAR K,K1,K2:INTEGER;
VAR CH:CHAR;

PROCEDURE UPDATEMASTER;
VAR CH:CHAR;
BEGIN
  CLRSCR;
  GOTOXY(3,3);WRITE('PASSING THE UPDATEMASTER PROCEDURE');
  REPEAT UNTIL KEYPRESSED;
  REPEAT READ(KBD,CH) UNTIL NOT KEYPRESSED;
END;

PROCEDURE SCANMASTER;
VAR CH:CHAR;
BEGIN
  CLRSCR;
  GOTOXY(3,3);WRITE('PASSING THE SCANMASTER PROCEDURE');
  REPEAT UNTIL KEYPRESSED;
  REPEAT READ(KBD,CH) UNTIL NOT KEYPRESSED;
END;

PROCEDURE MAKEREPORT;
VAR CH:CHAR;
BEGIN
  CLRSCR;
  GOTOXY(3,3);WRITE('PASSING THE MAKEREPORT PROCEDURE');
  REPEAT UNTIL KEYPRESSED;
  REPEAT READ(KBD,CH) UNTIL NOT KEYPRESSED;
END;

PROCEDURE MAKEBACKUP;
VAR CH:CHAR;
BEGIN
  CLRSCR;
  GOTOXY(3,3);WRITE('PASSING THE MAKEBACKUP PROCEDURE');
  REPEAT UNTIL KEYPRESSED;
  REPEAT READ(KBD,CH) UNTIL NOT KEYPRESSED;
END;

PROCEDURE EXTRACTRELOAD;
VAR CH:CHAR;
BEGIN
  CLRSCR;
  GOTOXY(3,3);WRITE('PASSING THE EXTRACTRELOAD PROCEDURE');
  REPEAT UNTIL KEYPRESSED;
  REPEAT READ(KBD,CH) UNTIL NOT KEYPRESSED;
END;
```

```
BEGIN

MAINLOOP:
  CLRSCR;
  FOR K1:=2 TO 79 DO
   BEGIN
   GOTOXY(K1,1);WRITE('*');
   GOTOXY(K1,25);WRITE('*');
   END;
  FOR K2:=2 TO 24 DO
   BEGIN
   GOTOXY(1,K2);WRITE('*');
   GOTOXY(80,K2);WRITE('*');
   END;
  GOTOXY(3,3);WRITE(' TP1S01:MAIN PROGRAM  ');
  GOTOXY(10,5);WRITE(' 1.  UPDATE MASTERFILE  ');
  GOTOXY(10,7);WRITE(' 2.  SCAN   MASTERFILE  ');
  GOTOXY(10,9);WRITE(' 3.  MAKE   REPORT      ');
  GOTOXY(10,11);WRITE(' 4.  MAKE   BACK-UP     ');
  GOTOXY(10,13);WRITE(' 5.  EXTRACT/RELOAD MASTER ');
  GOTOXY(10,15);WRITE(' 9.  END JOB            ');
  REPEAT
    GOTOXY(3,23);WRITE('                        ');
    GOTOXY(3,24);WRITE('                        ');
    GOTOXY(3,23);WRITE(' YOUR SELECTION-->    ');
    GOTOXY(3,24);WRITE('         < 1-5 OR 9 >   ');
    READ(KBD,CH);
    UNTIL (CH='1') OR (CH='2') OR (CH='3') OR
          (CH='4') OR (CH='5') OR (CH='9');
  IF CH='1' THEN UPDATEMASTER;
  IF CH='2' THEN SCANMASTER;
  IF CH='3' THEN MAKEREPORT;
  IF CH='4' THEN MAKEBACKUP;
  IF CH='5' THEN EXTRACTRELOAD;
  IF CH='9' THEN GOTO EOJ;
  GOTO MAINLOOP;

EOJ:
  GOTOXY(50,24);WRITE('END OF JOB');
  END.
```

Field Name	Description
. STATUS	สถานะของเรคอร์ด, เป็นตัวเลข Integer ø = ว่าง 1 = มีข้อมูลและใช้งานอยู่ 2 = เลิกใช้แล้ว
. TRUSTNAME	ชื่อของทรัสต์, เป็น String ยาว 40 ตัวอักษร
. PHONENUMBER	หมายเลขโทรศัพท์ของทรัสต์, เป็น String ยาว 7 ตัวอักษร
. AMOUNT	จำนวนเงินฝาก, Real Number
. RATE	อัตราดอกเบี้ย, Real Number
. STARTDAY	วันฝาก, เป็นตัวเลข integer, ใช้ 2 หลัก
. STARTMONTH	เดือนฝาก, เป็นตัวเลข Integer, ใช้ 2 หลัก
. STARTYEAR	ปีฝาก, เป็นตัวเลข Integer, ใช้ 2 หลัก
. ENDDAY	วันครบอายุ, เป็นตัวเลข Integer, ใช้ 2 หลัก
0. ENDMONTH	เดือนครบอายุ, เป็นตัวเลข Integer, ใช้ 2 หลัก
1. ENDYEAR	ปีครบอายุ, เป็นตัวเลข Integer, ใช้ 2 หลัก
2. PNNUMBER	หมายเลขตั๋วเงิน, เป็น String ยาว 10 ตัวอักษร
3. REFNO	หมายเลขประจำแผ่นเอกสาร, เป็น String ยาว 10 ตัวอักษร
4. REMARK	หมายเหตุ, เป็น String ยาว 40 ตัวอักษร

ที่ 2 ตารางแสดงลักษณะของเรคอร์ด

ของตั๋วสัญญาใช้เงินที่เราได้มาจากการเอาเงินไปฝากทรัสต์ ระบบนี้เป็นระบบงานเล็ก ๆ ที่ใคร ๆ ที่มีเงินฝากทรัสต์ไว้กินดอกเบี้ยก็สามารถจะนำไปใช้ได้ เราจะสมมติว่าผู้ใช้ระบบมีเงินฝากไว้ 2-3 ทรัสต์ และใช้บริการกันมานานพอสมควร จำนวนตั๋วที่มีอยู่ในมือจะเป็นจำนวนไม่เกิน 200 ฉบับ มาสเตอร์ไฟล์ของเราก็จะเตรียมเนื้อที่เอาไว้ 200 เรคอร์ด เราจะเคลียร์พื้นที่เอาไว้ด้วยโปรแกรมอันหนึ่ง โดยจะเซตค่าในฟิลด์ต่าง ๆ ให้เป็นศูนย์ (Zero) หรือให้เป็นช่องว่าง (space) ทั้งนี้แล้วแต่ว่าฟิลด์นั้นมีลักษณะเป็นตัวเลขหรือตัวหนังสือ ลักษณะของเรคอร์ดดังแสดงในรูปที่ 2 และโปรแกรมที่ใช้เคลียร์พื้นที่ดังแสดงในรูปที่ 3 ท่านผู้อ่านควรสังเกตว่าผมไม่เอาโปรแกรมเคลียร์พื้นที่เข้าไปรวมอยู่ในเมนโปรแกรม ทั้งนี้เพราะต้องการป้องกันความผิดพลาดในการใช้เมนโปรแกรม ถ้าเอามารวมกันแล้วเกิดผู้ใช้ไปเลือกการเคลียร์พื้นที่โดยไม่ได้ตั้งใจ ตัวมาสเตอร์ไฟล์ก็จะหายหมดได้

โอเค สมมติว่าเราวิ่งโปรแกรมเพื่อสร้างและเคลียร์พื้นที่ของมาสเตอร์ไฟล์เอาไว้แล้ว ตัวมาสเตอร์ไฟล์นี้ชื่อว่า "TP1DO1.DAT" ทีนี้ก็จะมาถึงขั้นทำรายละเอียดของโพรซีเดอร์ SCANMASTER โพรซีเดอร์นี้จะเริ่มจากเรคอร์ดที่ 1 ของมาสเตอร์ไฟล์ มันจะไปอ่านเอาเรคอร์ดดังกล่าวขึ้นมา นำเอาข้อมูลในฟิลด์ต่าง ๆ มาจัดเป็นรูปรายงานบนหน้าจอภาพ ทำเสร็จแล้วก็จะไปรอรับคำสั่ง ผู้ใช้สามารถจะเลือกที่จะกดแป้นต่าง ๆ ได้ดังนี้คือ กดแป้นอักษร F เพื่อเลื่อนไปข้างหน้า คือไปอ่านเอาเรคอร์ดถัดไปขึ้นมาแสดง กดแป้นอักษร B เพื่อย้อนหลัง คืออ่านเอาเรคอร์ดก่อนหน้านี้ไป 1 เรคอร์ดขึ้นมาแสดง ในกรณี

```
PROGRAM TP1S02;
LABEL MAINLOOP,EOJ;
TYPE  MASTERRECORD=RECORD
      STATUS:INTEGER;
      TRUSTNAME:STRING[40];
      PHONENUMBER:STRING[7];
      AMOUNT,RATE:REAL;
      STARTDAY,STARTMONTH,STARTYEAR:INTEGER;
      ENDDAY,ENDMONTH,ENDYEAR:INTEGER;
      PNNUMBER:STRING[10];
      REFNO:STRING[10];
      REMARK:STRING[60];
      END;
VAR MASTERFILE:FILE OF MASTERRECORD;
VAR MASTERBUFF:MASTERRECORD;
VAR K,K1,K2:INTEGER;
VAR CH:CHAR;

BEGIN
  CLRSCR;
  FOR K1:=2 TO 79 DO
   BEGIN
   GOTOXY(K1,1);WRITE('*');
   GOTOXY(K1,25);WRITE('*');
   END;
  FOR K2:=2 TO 24 DO
   BEGIN
   GOTOXY(1,K2);WRITE('*');
   GOTOXY(80,K2);WRITE('*');
   END;
  GOTOXY(20,1);WRITE(' TP1S02:CREATE MASTERFILE ');
  REPEAT
    GOTOXY(2,23);WRITE('                                                ');
    GOTOXY(2,24);WRITE('                                                ');
    GOTOXY(4,23);WRITE('ARE YOU SURE? OLD FILE WILL BE WIPED OUT');
    GOTOXY(4,24);WRITE('          Y=YES,  N=CANCEL  ');
    READ(KBD,CH);
    UNTIL (CH='Y') OR (CH='N');
  IF CH='N' THEN GOTO EOJ;

MAINLOOP:
  MASTERBUFF.STATUS:=0;
  MASTERBUFF.TRUSTNAME:='                                        ';
  MASTERBUFF.PHONENUMBER:='       ';
  MASTERBUFF.AMOUNT:=0.0;
  MASTERBUFF.RATE:=0.0;
  MASTERBUFF.STARTDAY:=0;
  MASTERBUFF.STARTMONTH:=0;
  MASTERBUFF.STARTYEAR:=0;
  MASTERBUFF.ENDDAY:=0;
  MASTERBUFF.ENDMONTH:=0;
  MASTERBUFF.ENDYEAR:=0;
  MASTERBUFF.PNNUMBER:='          ';
  MASTERBUFF.REFNO:='          ';
  MASTERBUFF.REMARK:='                                        ';
  ASSIGN(MASTERFILE,'TP1D01.DAT');
  REWRITE(MASTERFILE);
  FOR K:=1 TO 200 DO
    BEGIN
    SEEK(MASTERFILE,K);
    WRITE(MASTERFILE,MASTERBUFF);
    END;
  CLOSE(MASTERFILE);

EOJ:
  GOTOXY(50,24);WRITE('END OF JOB');
  END.
```

รูปที่ 3 โปรแกรมเพื่อสร้างมาสเตอร์ไฟล์

ที่อยู่เรคอร์ดที่ 1 อยู่แล้ว แล้วจะขอย้อนหลัง โปรแกรมจะต้องออกเสียงเตือน | และถ้าอยู่ที่เรคอร์ดที่ 200 อยู่แล้ว แล้วขอเลื่อนไปข้างหน้าก็จะต้องออกเสียง

รูปที่ 4 โปรแกรมที่ทำจนครบทุกโพรซีเดอร์แล้ว

```
PROGRAM TP1S07;
LABEL MAINLOOP,EOJ;
TYPE  MASTERRECORD=RECORD
      STATUS:INTEGER;
      TRUSTNAME:STRING[40];
      PHONENUMBER:STRING[7];
      AMOUNT,RATE:REAL;
      STARTDAY,STARTMONTH,STARTYEAR:INTEGER;
      ENDDAY,ENDMONTH,ENDYEAR:INTEGER;
```

เตือนเช่นกัน แป้นสุดท้ายที่จะเลือกได้ คือ E ซึ่งหมายถึง EXIT คือย้อนกลับไปสู่เมนูใหญ่

โปรแกรม TP1SØ3.PAS เป็นโปรแกรมหลังจากที่เราเติมโพรซีเดอร์ SCANMASTER เข้าไปแล้ว (โปรดดูหมายเหตุท้ายบทความ) โพรซีเดอร์นี้ไม่ยาวนัก เราสามารถทำได้โดยไม่ต้องแยกเป็นโพรซีเดอร์ย่อย และเนื่องจากสิ่งที่เราเติมเข้ามานี้มีอยู่ไม่กี่บรรทัด การดีบักจึงทำได้โดยง่าย ท่านไม่ต้องไปเขียนเป็นโฟลว์ชาร์ตหรือซูโดโค้ด ท่านไปนั่งที่หน้าเครื่องแล้วคีย์โปรแกรมเข้าไปได้เลย โพรซีเดอร์ SCANMASTER นี้จะใช้เวลาไม่เกินครึ่งชั่วโมงก็เสร็จ เสร็จในที่นี้ผมหมายถึงทั้งการคิด การเขียน การคอมพายล์ และการทดสอบว่ามันทำงานได้จริง ๆ ในเรื่องความเร็วของการทำงานนั้น เครื่องทำได้เร็วเกินพอ

เสร็จจากการเติมโพรซีเดอร์ SCANMASTER แล้วเราก็อาจมารายละเอียดของการอัพเดต เรื่องนี้เราอาจแบ่งออกเป็นโพรซีเดอร์ย่อย 3 โพรซีเดอร์ คือ ADDRECORD, CHANGERECORD และ DELETE RECORD การ ADDRECORD จะเกิดขึ้นเมื่อไปฝากเงินครั้งใหม่ ซึ่งเราจะได้ตั๋วสัญญาใช้เงินมาใหม่อีก 1 ฉบับ เราต้องนำมาบันทึกข้อมูลเข้าไปในมาสเตอร์ไฟล์ โปรแกรมจะต้องวิ่งหาเรคอร์ดที่ยังว่างเปล่ามาใช้ มันจะค้นหาตั้งแต่เรคอร์ดที่ 1 เป็นต้นไป เจออันไหนที่มีรหัส "STATUS" เป็นเลข 0 ก็ใช้เรคอร์ดนั้น ถ้าหาไม่ได้ก็จะออกคำเตือนว่าให้ทำ EXTRACT และ RELOAD เสียก่อน หรือไม่ก็ต้องขยายขนาดของมาสเตอร์ไฟล์ แต่ถ้าหาได้มันก็จะพร้อมที่จะให้ท่านคีย์ข้อมูลจากตั๋วฉบับใหม่เข้าเครื่อง ในตัวอย่างนี้ผมจะพยายามที่

ให้ดูง่าย ๆ คือใช้วิธีโต้ตอบทีละบรรทัด เมื่อได้ครบแล้วก็จะจัดรูปเสียใหม่ให้มีหน้าตาเหมือนตอนทำ SCANMASTER ทั้งนี้เพื่อให้ผู้ใช้เครื่องตรวจดูข้อมูลเสียก่อน ถึงจุดนั้นโปรแกรมจะรอรับคำตอบ ถ้าท่านกดปุ่ม Y ก็หมายความว่าข้อมูลถูกต้อง โปรแกรมก็จะนำไปอัพเดตลงจานแม่เหล็กและหลุดออกจากโพรซีเดอร์นี้ ถ้าท่านกดปุ่ม N ก็หมายความว่าข้อมูลยังไม่ถูกต้อง ซึ่งโปรแกรมจะกลับไปเริ่มรับข้อมูลกันใหม่ และถ้าท่านกดปุ่ม C ก็จะหมายความว่าท่านเปลี่ยนใจ (cancel) คือจะไม่มีการอัพเดตแล้วโปรแกรมก็จะหลุดออกจากโพรซีเดอร์นี้เช่นกัน

จากที่วางแผนเอาไว้ข้างบนนี้เราจะเห็นว่า โพรซีเดอร์ ADDRECORD อันเดียวก็ยาวพอสมควรแล้ว ดังนั้นเราจึงควรทำแค่ ADDRECORD เสียก่อน เขียนและทดสอบให้ดีเสียก่อนที่จะไปทำรายละเอียดในโพรซีเดอร์อื่น โปรแกรมที่ TP1S04.PAS เป็นโปรแกรมที่ทำรายละเอียดของ ADDRECORD เสร็จแล้ว โปรแกรมส่วนนี้ผมก็เชื่อว่าท่านจะทำเสร็จได้ในไม่เกินครึ่งชั่วโมงเช่นกัน

เขียนมาถึงตอนนี้ผมก็อยากจะชี้ให้เห็นข้อดีข้อหนึ่งของการทำงานแบบ "ค่อยทำค่อยไป" คือ ในตอนที่เราทดสอบโพรซีเดอร์ ADDRECORD นี้เราสามารถใช้โพรซีเดอร์ SCANMASTER เข้าช่วย ทั้งนี้โดยทดลองเติมข้อมูลเข้าไปในมาสเตอร์ไฟล์แล้วก็ถอยกลับมาสู่เมนเมนู แล้วเลือกทำ SCANMASTER ดู เราจะตรวจดูได้ว่าข้อมูลถูกเก็บเข้ามาสเตอร์ไฟล์ได้จริงหรือไม่ เก็บครบถ้วนหรือไม่ และอยู่ที่เรคอร์ดแรกที่เป็นที่ว่างจริงหรือไม่ ฯลฯ นี่เป็นการใช้ประโยชน์จากส่วนของโปรแกรม

```
      PNNUMBER:STRING[10];
      REFNO:STRING[10];
      REMARK:STRING[60];
      END;
VAR MASTERFILE:FILE OF MASTERRECORD;
VAR BACKUPFILE:FILE OF MASTERRECORD;
VAR EXTRACTEDFILE:FILE OF MASTERRECORD;
VAR MASTERBUFF:MASTERRECORD;
VAR CH:CHAR;

PROCEDURE BEEP;
BEGIN
  SOUND(400);DELAY(50);
  NOSOUND;DELAY(50);
END;

PROCEDURE CHECKDATE(DAY,MONTH,YEAR:INTEGER;VAR ERRSWT:INTEGER);
LABEL E1;
BEGIN
  ERRSWT:=0;
  IF (YEAR<20) OR (YEAR>99) THEN
    BEGIN ERRSWT:=1;GOTO E1;END;
  IF (MONTH<1) OR (MONTH>12) THEN
    BEGIN ERRSWT:=1;GOTO E1;END;
  IF (DAY<1) OR (DAY>31) THEN
    BEGIN ERRSWT:=1;GOTO E1;END;
  IF ((MONTH=4) OR (MONTH=6) OR (MONTH=9) OR (MONTH=11)) AND
      (DAY>30) THEN BEGIN ERRSWT:=1;GOTO E1;END;
  IF (MONTH=2) AND ((YEAR MOD 4)=3) AND (DAY>29) THEN
    BEGIN ERRSWT:=1;GOTO E1;END;
  IF (MONTH=2) AND ((YEAR MOD 4)<>3) AND (DAY>28) THEN
    BEGIN ERRSWT:=1;GOTO E1;END;
E1:
END;

PROCEDURE NEWSCREEN;
VAR K1,K2:INTEGER;
BEGIN
  CLRSCR;
  FOR K1:=2 TO 79 DO
   BEGIN
   GOTOXY(K1,1);WRITE('*');
   GOTOXY(K1,25);WRITE('*');
   END;
  FOR K2:=2 TO 24 DO
   BEGIN
   GOTOXY(1,K2);WRITE('*');
   GOTOXY(80,K2);WRITE('*');
   END;
END;

PROCEDURE PRINTMASTER(RECNO:INTEGER);
BEGIN
  GOTOXY(3,3);WRITE('RECORD NO : ',RECNO);
  GOTOXY(3,4);WRITE('RECORD STATUS : ',MASTERBUFF.STATUS);
  GOTOXY(20,7);WRITE('TRUST-NAME : ',MASTERBUFF.TRUSTNAME);
  GOTOXY(20,8);WRITE('  PHONE-NO : ',MASTERBUFF.PHONENUMBER);
  GOTOXY(4,10);WRITE('PN-NUMBER : ',MASTERBUFF.PNNUMBER);
  GOTOXY(50,10);WRITE('DATE : ',MASTERBUFF.STARTDAY:2,' ',
                MASTERBUFF.STARTMONTH:2,' ',MASTERBUFF.STARTYEAR:2);
  GOTOXY(10,12);WRITE('AMOUNT : ',MASTERBUFF.AMOUNT:12:2);
  GOTOXY(10,14);WRITE('RATE : ',MASTERBUFF.RATE:8:4);
  GOTOXY(10,16);WRITE('DUE-DATE : ',MASTERBUFF.ENDDAY:2,' ',
                MASTERBUFF.ENDMONTH:2,' ',MASTERBUFF.ENDYEAR:2);
  GOTOXY(10,18);WRITE('REMARK : ');
  GOTOXY(10,19);WRITE(MASTERBUFF.REMARK);
END;

PROCEDURE ADDRECORD;
LABEL L1,L2,L3,L4;
VAR RECNO,K,LN,ERRSWT:INTEGER;
VAR CH:CHAR;
BEGIN
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : ADD RECORD  ');
  ASSIGN(MASTERFILE,'TP1D01.DAT');
  RESET(MASTERFILE);
  RECNO:=1;
L1:
  SEEK(MASTERFILE,RECNO);
  READ(MASTERFILE,MASTERBUFF);
  IF MASTERBUFF.STATUS=0 THEN GOTO L2;
  IF RECNO=200 THEN
    BEGIN
    FOR K:=1 TO 5 DO
      BEGIN
      GOTOXY(3,23);WRITE('FILE IS FULL, PLEASE DO EXTRACT/RELOAD');
      SOUND(800);DELAY(50);NOSOUND;DELAY(50);
      GOTOXY(3,23);WRITE('                                        ');
      END;
    GOTO L4;
    END;
```

```
  RECNO:=RECNO+1;
  GOTO L1;
L2:
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : ADD RECORD  ');
  GOTOXY(3,3);WRITE('RECORD NO : ',RECNO,' IS AVAILABLE');
  GOTOXY(3,4);WRITE('PLEASE ENTER THE DATA');
  GOTOXY(5,6);WRITE(' TRUST-NAME= ');READ(MASTERBUFF.TRUSTNAME);
  GOTOXY(5,7);WRITE('    PHONE-NO= ');READ(MASTERBUFF.PHONENUMBER);
  GOTOXY(5,8);WRITE('  PN-NUMBER= ');READ(MASTERBUFF.PNNUMBER);
  GOTOXY(5,9);WRITE('      REF-NO= ');READ(MASTERBUFF.REFNO);
  REPEAT
    GOTOXY(5,10);WRITE('                              ');
    GOTOXY(5,11);WRITE('                              ');
    GOTOXY(5,12);WRITE('                              ');
    GOTOXY(5,10);WRITE('  START-DAY= ');READ(MASTERBUFF.STARTDAY);
    GOTOXY(5,11);WRITE('START-MONTH= ');READ(MASTERBUFF.STARTMONTH);
    GOTOXY(5,12);WRITE(' START-YEAR= ');READ(MASTERBUFF.STARTYEAR);
    CHECKDATE(MASTERBUFF.STARTDAY,MASTERBUFF.STARTMONTH,
          MASTERBUFF.STARTYEAR,ERRSWT);
    IF ERRSWT=1 THEN BEGIN BEEP;BEEP;END;
    UNTIL ERRSWT=0;
  GOTOXY(5,13);WRITE('      AMOUNT= ');READ(MASTERBUFF.AMOUNT);
  GOTOXY(5,14);WRITE('        RATE= ');READ(MASTERBUFF.RATE);
  REPEAT
    GOTOXY(5,15);WRITE('                              ');
    GOTOXY(5,16);WRITE('                              ');
    GOTOXY(5,17);WRITE('                              ');
    GOTOXY(5,15);WRITE('    DUE-DAY= ');READ(MASTERBUFF.ENDDAY);
    GOTOXY(5,16);WRITE('  DUE-MONTH= ');READ(MASTERBUFF.ENDMONTH);
    GOTOXY(5,17);WRITE('   DUE-YEAR= ');READ(MASTERBUFF.ENDYEAR);
    CHECKDATE(MASTERBUFF.ENDDAY,MASTERBUFF.ENDMONTH,
          MASTERBUFF.ENDYEAR,ERRSWT);
    IF ERRSWT=1 THEN BEGIN BEEP;BEEP;END;
    UNTIL ERRSWT=0;
  GOTOXY(5,18);WRITE('      REMARK= ');READ(MASTERBUFF.REMARK);
  MASTERBUFF.STATUS:=1;
  LN:=LENGTH(MASTERBUFF.TRUSTNAME);
  FOR K:=LN TO 40 DO
    MASTERBUFF.TRUSTNAME:=MASTERBUFF.TRUSTNAME+' ';
  LN:=LENGTH(MASTERBUFF.PHONENUMBER);
  FOR K:=LN TO 7 DO
    MASTERBUFF.PHONENUMBER:=MASTERBUFF.PHONENUMBER+' ';
  LN:=LENGTH(MASTERBUFF.PNNUMBER);
  FOR K:=LN TO 10 DO
    MASTERBUFF.PNNUMBER:=MASTERBUFF.PNNUMBER+' ';
  LN:=LENGTH(MASTERBUFF.REFNO);
  FOR K:=LN TO 10 DO
    MASTERBUFF.REFNO:=MASTERBUFF.REFNO+' ';
  LN:=LENGTH(MASTERBUFF.REMARK);
  FOR K:=LN TO 60 DO
    MASTERBUFF.REMARK:=MASTERBUFF.REMARK+' ';
L3:
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : ADD RECORD  ');
  PRINTMASTER(RECNO);
  GOTOXY(3,23);WRITE('IS IT OK?                ');
  GOTOXY(3,24);WRITE('   Y=YES, N=NO, C=CANCEL ');
  READ(KBD,CH);
  IF (CH='Y') THEN
     BEGIN
     SEEK(MASTERFILE,RECNO);
     WRITE(MASTERFILE,MASTERBUFF);
     GOTO L4;
     END;
  IF (CH='N') THEN GOTO L2;
  IF (CH='C') THEN GOTO L4;
L4:
  CLOSE(MASTERFILE);
END;

PROCEDURE CHANGERECORD;
LABEL L1,L2,L3,L4,L5,L6,L8,L9;
VAR RECNO,K,LN,ERRSWT:INTEGER;
VAR PNNO:STRING[10];
VAR CH:CHAR;
BEGIN
  ASSIGN(MASTERFILE,'TP1D01.DAT');
  RESET(MASTERFILE);
L1:
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : CHANGE RECORD  ');
  GOTOXY(3,3);WRITE('WHAT PN-NUMBER? ');READ(PNNO);
  LN:=LENGTH(PNNO);
  FOR K:=LN TO 10 DO PNNO:=PNNO+' ';
  RECNO:=1;
  GOTOXY(3,5);WRITE('SEARCHING, PLEASE WAIT ');
L2:
  SEEK(MASTERFILE,RECNO);
  READ(MASTERFILE,MASTERBUFF);
  IF MASTERBUFF.PNNUMBER=PNNO THEN GOTO L3;
  IF RECNO=200 THEN
```

ที่เพิ่งทำเสร็จไป และในขณะเดียวกันมันยังเป็นการช่วยตอกย้ำว่า SCANMASTER นั้นเราทำไว้ถูกต้องดีแล้ว การที่เราสามารถสร้างความมั่นใจได้แบบเป็นลูกคลื่นที่ทะยอยไล่ตามกันมานี้จะทำให้ไม่เป็นกังวล คลื่นลูกแรก ๆ จะส่งผลช่วยเสริมคลื่นลูกใหม่ต่อ ๆ ไป มันจึงเป็นวิธีการที่ดียิ่ง

แต่ก่อนที่จะลงรายละเอียดของโพรซีเดอร์อื่นต่อไป ผมก็อยากจะขอให้ท่านศึกษาโปรแกรม TP1S04.PAS ซ้ำอีกสักหน่อย ในโปรแกรมนี้เราจะเห็นว่ามีสิ่งที่เราน่าจะปรับปรุงเสียก่อน 2-3 เรื่อง เรื่องแรกคือ การรับข้อมูลประเภทวัน-เดือน-ปีนั้นมี 2 ครั้ง และผมยังไม่ได้ทำการอีดิตติ้งเลย คือไม่ได้เช็คว่าวันที่กับปี พ.ศ.มันสอดคล้องกันหรือไม่ เช่น วันที่ 30 เดือนกุมภาพันธ์ นั้นเป็นไปไม่ได้ เรื่องนี้เราควรจะแยกเอามาทำเป็นโพรซีเดอร์พิเศษเอาไว้ เราอาจเรียกชื่อว่า CHECKDATE ทั้งนี้โดยต้องมีพารามิเตอร์ส่งผ่านเข้าจำนวน 3 ตัวซึ่งได้แก่ DAY, MONTH, YEAR และส่งผ่านออก 1 ตัว คือ ERRSWT ซึ่งเป็นผลของการตรวจสอบ โพรซีเดอร์นี้จะใช้ร่วมกัน เราจึงควรเอาไปใส่ไว้ตอนบนของเด็คคาเรชันพาร์ท (Declaration Part)

เรื่องที่ 2 ที่ควรปรับปรุงคือเรื่องการเอาข้อมูลจากเรคอร์ดของมาสเตอร์ไฟล์ขึ้นมาจัดแสดงบนจอภาพ เรื่องนี้มีปรากฏอยู่ 2 แห่งแล้วในโปรแกรมของเรา คือใน SCANMASTER ก็มี และใน ADDRECORD ก็มี และผมเชื่อว่าเรายังจะต้องใช้อีกในโพรซีเดอร์อื่น ๆ ที่ยังไม่ได้แตกรายละเอียด เรื่องนี้เราก็ควรนำมาทำเป็นโพรซีเดอร์ทั่วไปเพื่อให้ส่วนของโปรแกรมเรียกไปใช้งานได้เช่นกัน

นอกจากนี้เราก็ยังมีเรื่องการวาด

```
    BEGIN
    GOTOXY(3,7);WRITE('NOT IN MASTERFILE');
    SOUND(800);DELAY(50);NOSOUND;DELAY(50);
    GOTO L8;
    END;
  RECNO:=RECNO+1;
  GOTO L2;
L3:
  IF MASTERBUFF.STATUS<>1 THEN
    BEGIN
    GOTOXY(3,7);WRITE('STATUS IS NOT 1, IT CANNOT BE CHANGE');
    SOUND(800);DELAY(50);NOSOUND;DELAY(50);
    GOTO L8;
    END;
L4:
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : CHANGE RECORD  ');
  PRINTMASTER(RECNO);
  GOTOXY(3,23);WRITE('IS IT THE RECORD?           ');
  GOTOXY(3,24);WRITE('   Y=YES, N=NO, C=CANCEL ');
  READ(KBD,CH);
  IF (CH='Y') THEN GOTO L5;
  IF (CH='N') THEN GOTO L1;
  IF (CH='C') THEN GOTO L9;
L5:
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : CHANGE RECORD  ');
  GOTOXY(3,3);WRITE('RECORD NO : ',RECNO);
  GOTOXY(3,4);WRITE('PLEASE ENTER THE DATA');
  GOTOXY(5,6);WRITE(' TRUST-NAME=',MASTERBUFF.TRUSTNAME,' ');
  GOTOXY(17,6);READ(MASTERBUFF.TRUSTNAME);
  GOTOXY(5,7);WRITE('   PHONE-NO=',MASTERBUFF.PHONENUMBER,' ');
  GOTOXY(17,7);READ(MASTERBUFF.PHONENUMBER);
  GOTOXY(5,8);WRITE(' PN-NUMBER=',MASTERBUFF.PNNUMBER,'<CANNOT CHANGE>');
  GOTOXY(5,9);WRITE('     REF-NO=',MASTERBUFF.REFNO,' ');
  GOTOXY(17,9);READ(MASTERBUFF.REFNO);
  REPEAT
    GOTOXY(5,10);WRITE('                              ');
    GOTOXY(5,11);WRITE('                              ');
    GOTOXY(5,12);WRITE('                              ');
    GOTOXY(5,10);WRITE('  START-DAY=',MASTERBUFF.STARTDAY,' ');
    GOTOXY(17,10);READ(MASTERBUFF.STARTDAY);
    GOTOXY(5,11);WRITE('START-MONTH=',MASTERBUFF.STARTMONTH,' ');
    GOTOXY(17,11);READ(MASTERBUFF.STARTMONTH);
    GOTOXY(5,12);WRITE(' START-YEAR=',MASTERBUFF.STARTYEAR,' ');
    GOTOXY(17,12);READ(MASTERBUFF.STARTYEAR);
    CHECKDATE(MASTERBUFF.STARTDAY,MASTERBUFF.STARTMONTH,
          MASTERBUFF.STARTYEAR,ERRSWT);
    IF ERRSWT=1 THEN BEGIN BEEP;BEEP;END;
    UNTIL ERRSWT=0;
  GOTOXY(5,13);WRITE('     AMOUNT=',MASTERBUFF.AMOUNT:12:2,' ');
  GOTOXY(17,13);READ(MASTERBUFF.AMOUNT);
  GOTOXY(5,14);WRITE('       RATE=',MASTERBUFF.RATE:8:4,' ');
  GOTOXY(17,14);READ(MASTERBUFF.RATE);
  REPEAT
    GOTOXY(5,15);WRITE('                              ');
    GOTOXY(5,16);WRITE('                              ');
    GOTOXY(5,17);WRITE('                              ');
    GOTOXY(5,15);WRITE('    DUE-DAY=',MASTERBUFF.ENDDAY,' ');
    GOTOXY(17,15);READ(MASTERBUFF.ENDDAY);
    GOTOXY(5,16);WRITE('  DUE-MONTH=',MASTERBUFF.ENDMONTH,' ');
    GOTOXY(17,16);READ(MASTERBUFF.ENDMONTH);
    GOTOXY(5,17);WRITE('   DUE-YEAR=',MASTERBUFF.ENDYEAR,' ');
    GOTOXY(17,17);READ(MASTERBUFF.ENDYEAR);
    CHECKDATE(MASTERBUFF.ENDDAY,MASTERBUFF.ENDMONTH,
          MASTERBUFF.ENDYEAR,ERRSWT);
    IF ERRSWT=1 THEN BEGIN BEEP;BEEP;END;
    UNTIL ERRSWT=0;
  GOTOXY(5,18);WRITE('     REMARK=',MASTERBUFF.REMARK,' ');
  GOTOXY(17,18);READ(MASTERBUFF.REMARK);
  MASTERBUFF.STATUS:=1;
  LN:=LENGTH(MASTERBUFF.TRUSTNAME);
  FOR K:=LN TO 40 DO
    MASTERBUFF.TRUSTNAME:=MASTERBUFF.TRUSTNAME+' ';
  LN:=LENGTH(MASTERBUFF.PHONENUMBER);
  FOR K:=LN TO 7 DO
    MASTERBUFF.PHONENUMBER:=MASTERBUFF.PHONENUMBER+' ';
  LN:=LENGTH(MASTERBUFF.REFNO);
  FOR K:=LN TO 10 DO
    MASTERBUFF.REFNO:=MASTERBUFF.REFNO+' ';
  LN:=LENGTH(MASTERBUFF.REMARK);
  FOR K:=LN TO 60 DO
    MASTERBUFF.REMARK:=MASTERBUFF.REMARK+' ';
L6:
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : CHANGE RECORD  ');
  PRINTMASTER(RECNO);
  GOTOXY(3,23);WRITE('IS IT OK?                   ');
  GOTOXY(3,24);WRITE('   Y=YES, N=NO, C=CANCEL ');
  READ(KBD,CH);
  IF (CH='Y') THEN
    BEGIN
```

กรอบของจอภาพ สิ่งนี้ควรเอาไปประกาศเป็นโพรซีเดอร์เอาไว้ แล้วเรียกใช้ในเมนบล็อก ทั้งนี้เพื่อให้เมนบล็อกสั้นและอ่านเข้าใจง่ายขึ้น

จากข้อคิดต่าง ๆ ดังกล่าวมาข้างบนนี้ ผมก็ไปจัดการปรับปรุงโปรแกรมเสียก่อน ซึ่งได้ผลตามโปรแกรม TP1S ϕ5.PAS ท่านผู้อ่านควรสังเกตว่า โปรแกรมดูจะสั้นและกะทัดรัดดียิ่งขึ้น สิ่งต่าง ๆ เหล่านี้ทำได้เพราะภาษาปาสคาลเปิดโอกาสให้เราเขียนเป็นรูทีนหรือโพรซีเดอร์ได้หลาย ๆ รูปแบบ

ครับ ขอผ่านมาถึงการทำรายละเอียดของโพรซีเดอร์ CHANGERECORD การอัพเดตแบบนี้หมายถึงการแก้ไขข้อมูลในฟิลด์ต่าง ๆ ของเรคอร์ดของมาสเตอร์ไฟล์ แต่การแก้ไขนี้จะทำได้กับเรคอร์ดประเภทที่มี STATUS เป็นเลข 1 เท่านั้น และมันจะเปลี่ยน STATUS และ PNNUMBER ไม่ได้ การอัพเดทแบบนี้จำเป็นต้องมีเพราะเราอาจมาตรวจพบในภายหลังว่าข้อมูลที่เราใส่เข้าไปตอนทำเรื่อง ADDRECORD มีผิดพลาดอยู่บ้างในบางฟิลด์ การแก้ไขแบบนี้ต้องควบคุม ในงานจริง ๆ นั้นต้องมีการบันทึกเหตุการณ์เอาไว้คือต้องรู้ว่าใครทำ, ทำเมื่อไหร่, มีเหตุผลอย่างไร สำหรับตัวอย่างของเราในครั้งนี้ผมจะทำอย่างย่นย่อ คือไม่มีการจดบันทึกอะไรไว้ เพราะถ้าใส่รายละเอียดมากก็อาจทำให้ไขว้เขวเปล่า ๆ โพรซีเดอร์นี้จะทำงานคล้าย ๆ กับ ADDRECORD แต่จะมีการเอาข้อมูลเก่ามาแสดงเสียก่อนที่จะรับข้อมูลใหม่ และเมื่อผ่านไปครบทุกฟิลด์แล้วก็จะให้ผู้ใช้ตรวจสอบอีกครั้งหนึ่งก่อนที่จะนำข้อมูลไปอัพเดตจริง ๆ โพรซีเดอร์นี้จะยุ่งยากกว่าโพรซีเดอร์ ADDRECORD อยู่เล็กน้อยคือ ต้องตีความให้ดี ๆ ว่าอะไร

```
    SEEK(MASTERFILE,RECNO);
    WRITE(MASTERFILE,MASTERBUFF);
    GOTO L9;
    END;
  IF (CH='N') THEN GOTO L6;
  IF (CH='C') THEN GOTO L9;
L8:
  GOTOXY(3,9);WRITE('HIT ANY KEY TO RETURN ');
  REPEAT UNTIL KEYPRESSED;
  REPEAT READ(KBD,CH) UNTIL NOT KEYPRESSED;
L9:
  CLOSE(MASTERFILE);
END;

PROCEDURE DELETERECORD;
LABEL L1,L2,L3,L4,L8,L9;
VAR RECNO,K,LN:INTEGER;
VAR PNNO:STRING[10];
VAR CH:CHAR;
BEGIN
  ASSIGN(MASTERFILE,'TP1D01.DAT');
  RESET(MASTERFILE);
L1:
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : DELETE RECORD  ');
  GOTOXY(3,3);WRITE('WHAT PN-NUMBER? ');READ(PNNO);
  LN:=LENGTH(PNNO);
  FOR K:=LN TO 10 DO PNNO:=PNNO+' ';
  RECNO:=1;
  GOTOXY(3,5);WRITE('SEARCHING, PLEASE WAIT ');
L2:
  SEEK(MASTERFILE,RECNO);
  READ(MASTERFILE,MASTERBUFF);
  IF MASTERBUFF.PNNUMBER=PNNO THEN GOTO L3;
  IF RECNO=200 THEN
    BEGIN
    GOTOXY(3,7);WRITE('NOT IN MASTERFILE');
    SOUND(800);DELAY(50);NOSOUND;DELAY(50);
    GOTO L8;
    END;
  RECNO:=RECNO+1;
  GOTO L2;
L3:
  IF MASTERBUFF.STATUS<>1 THEN
    BEGIN
    GOTOXY(3,7);WRITE('STATUS IS NOT 1, IT CANNOT BE DELETED');
    SOUND(800);DELAY(50);NOSOUND;DELAY(50);
    GOTO L8;
    END;
L4:
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : DELETE RECORD  ');
  PRINTMASTER(RECNO);
  GOTOXY(3,23);WRITE('IS IT THE RECORD?           ');
  GOTOXY(3,24);WRITE('   Y=YES, N=NO, C=CANCEL ');
  READ(KBD,CH);
  IF (CH='Y') THEN
    BEGIN
    MASTERBUFF.STATUS:=2;
    SEEK(MASTERFILE,RECNO);
    WRITE(MASTERFILE,MASTERBUFF);
    GOTO L9;
    END;
  IF (CH='N') THEN GOTO L1;
  IF (CH='C') THEN GOTO L9;
L8:
  GOTOXY(3,9);WRITE('HIT ANY KEY TO RETURN ');
  REPEAT UNTIL KEYPRESSED;
  REPEAT READ(KBD,CH) UNTIL NOT KEYPRESSED;
L9:
  CLOSE(MASTERFILE);
END;

PROCEDURE UPDATEMASTER;
VAR CH:CHAR;
BEGIN
  REPEAT
    NEWSCREEN;
    GOTOXY(20,1);WRITE(' PROCEDURE : UPDATE MASTER  ');
    GOTOXY(10,5);WRITE(' 1.  ADD RECORD      ');
    GOTOXY(10,7);WRITE(' 2.  CHANGE RECORD   ');
    GOTOXY(10,9);WRITE(' 3.  DELETE RECORD   ');
    GOTOXY(10,11);WRITE(' 9.  RETURN         ');
    REPEAT
      GOTOXY(3,23);WRITE('                        ');
      GOTOXY(3,24);WRITE('                        ');
      GOTOXY(3,23);WRITE(' YOUR SELECTION-->    ');
      GOTOXY(3,24);WRITE('       <1-3 OR 9>     ');
      READ(KBD,CH);
      UNTIL (CH='1') OR (CH='2') OR (CH='3') OR (CH='9');
    IF CH='1' THEN ADDRECORD;
```

เป็นข้อมูลเก่า (OLDDATA) และอะไร
เป็นข้อมูลใหม่ (NEWDATA) ในโปร-
แกรมตัวอย่างอันนี้สิ่งที่คีย์เข้าไปนั้นจะ
นำไปทับสิ่งที่มีอยู่ในเรกอร์ดที่อยู่ใน
เมมโมรีทันที ดังนั้นข้อมูลเก่าดั้งเดิมจะ
หายไปแล้ว ถ้าท่านอยากได้ข้อมูลเก่า
จริง ๆ ก็จะต้องยกเลิกการอัพเดทเสีย
ก่อน แล้วย้อนกลับมาใหม่อีกครั้งหนึ่ง
จากเมนเมนู

TP1S06.PAS เป็นโปรแกรมที่เรา
เติมรายละเอียดของโพรซีเดอร์เรื่อง
CHANGERECORD แล้ว การเติม
ส่วนนี้เข้ามาผมใช้เวลาประมาณ 1 ชั่วโม

จากโปรแกรมTP1S06.PAS ผม
คิดว่า ท่านผู้อ่านก็คงจะทำต่อไปจนจบได้
คือใส่รายละเอียดของทุก ๆ โพรซีเดอร์ได้
ผมจะไม่ขอพูดถึงรายละเอียดเหล่านี้
แล้ว เพราะเดี๋ยวจะเข้าใจผิดว่าบทความ
นี้เป็นเรื่องการพัฒนาระบบ "Promis
sory Note Recording System" แต่
อย่างไรก็ตาม ผมก็อยากจะให้ท่านได้
เห็นภาพอันสุดท้ายเอาไว้ด้วย รูปที่
เป็นโปรแกรมที่ทำจนสำเร็จเรียบร้อย
แล้ว โปรแกรมนี้อาจยังไม่สมบูรณ์ดีนัก
แต่ผมเชื่อว่ามันวิ่งได้และมีความเชื่อถือ
ได้ 100% คือจะไม่ทำให้มาสเตอร์ไฟล์
ของเราเสียหาย การอัพเดตจะถูกต้อง
ตรงจุด ไม่มีการทำงานเลอะเทอะ และ
ถ้าปรับปรุงวิธีการรับข้อมูลตัวเลขให้
ดีขึ้นอีกนิดหน่อย เช่น ใช้ READ
(KBD, Var) แทนการ READ ธรรมด
โปรแกรมนี้จะวิ่งได้ทุกครั้งโดยไม่มีกา
ตายกลางอากาศ (Abnormal Termi
nation)

มาวิเคราะห์ข้อดีข้อเสีย กันเสียหน่อย

ถ้ามองอย่างผิวเผินเราจะเห็นข้อ

```
    IF CH='2' THEN CHANGERECORD;
    IF CH='3' THEN DELETERECORD;
    UNTIL CH='9';
END;

PROCEDURE SCANMASTER;
LABEL L1,L2;
VAR RECNO:INTEGER;
VAR CH:CHAR;
BEGIN
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : SCAN MASTER  ');
  ASSIGN(MASTERFILE,'TP1D01.DAT');
  RESET(MASTERFILE);
  RECNO:=1;
L1:
  SEEK(MASTERFILE,RECNO);
  READ(MASTERFILE,MASTERBUFF);
  PRINTMASTER(RECNO);
  GOTOXY(3,23);WRITE('OPTION : F=FOREWARD, B=BACKWARD, E=EXIT ');
  READ(KBD,CH);
  IF (CH='F') AND (RECNO<200) THEN
     BEGIN RECNO:=RECNO+1;GOTO L1;END;
  IF (CH='B') AND (RECNO>1) THEN
     BEGIN RECNO:=RECNO-1;GOTO L1;END;
  IF (CH='E') THEN GOTO L2;
  BEEP;BEEP;
  GOTO L1;
L2:
  CLOSE(MASTERFILE);
END;

PROCEDURE MAKEREPORT;
VAR CH:CHAR;
BEGIN
  CLRSCR;
  GOTOXY(3,3);WRITE('PASSING THE MAKEREPORT PROCEDURE');
  REPEAT UNTIL KEYPRESSED;
  REPEAT READ(KBD,CH) UNTIL NOT KEYPRESSED;
END;

PROCEDURE MAKEBACKUP;
VAR RECNO:INTEGER;
VAR CH:CHAR;
BEGIN
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : MAKE BACK-UP  ');
  GOTOXY(3,3);WRITE(' MAKING BACK-UP, PLEASE WAIT  ');
  ASSIGN(MASTERFILE,'TP1D01.DAT');
  RESET(MASTERFILE);
  ASSIGN(BACKUPFILE,'TP1D02.DAT');
  REWRITE(BACKUPFILE);
  FOR RECNO:=1 TO 200 DO
    BEGIN
    SEEK(MASTERFILE,RECNO);
    READ(MASTERFILE,MASTERBUFF);
    SEEK(BACKUPFILE,RECNO);
    WRITE(BACKUPFILE,MASTERBUFF);
    END;
  CLOSE(MASTERFILE);
  CLOSE(BACKUPFILE);
  GOTOXY(3,4);WRITE(' COMPLETED  ');
  DELAY(1000);
END;

PROCEDURE EXTRACTRELOAD;
LABEL E1;
VAR MASTERRECNO,EXTRACTEDRECNO:INTEGER;
VAR CH:CHAR;
BEGIN
  NEWSCREEN;
  GOTOXY(20,1);WRITE(' PROCEDURE : EXTRACT-RELOAD  ');
  GOTOXY(3,3);WRITE('ARE YOU SURE? ALL RECORDS TYPE "2" WILL BE DESTROYED');
  GOTOXY(3,4);WRITE('      Y=YES, C=CANCEL    ');
  REPEAT
    READ(KBD,CH);
    UNTIL (CH='Y') OR (CH='C');
  IF CH='C' THEN GOTO E1;
  GOTOXY(3,6);WRITE(' DOING EXTRACT, PLEASE WAIT  ');
  ASSIGN(MASTERFILE,'TP1D01.DAT');
  RESET(MASTERFILE);
  ASSIGN(EXTRACTEDFILE,'TP1D03.DAT');
  REWRITE(EXTRACTEDFILE);
  EXTRACTEDRECNO:=0;
  FOR MASTERRECNO:=1 TO 200 DO
    BEGIN
    SEEK(MASTERFILE,MASTERRECNO);
    READ(MASTERFILE,MASTERBUFF);
    IF MASTERBUFF.STATUS=1 THEN
      BEGIN
      EXTRACTEDRECNO:=EXTRACTEDRECNO+1;
      SEEK(EXTRACTEDFILE,EXTRACTEDRECNO);
```

3 อย่าง คือ (ก) เขียนทีละส่วน ทำให้จำกัดปัญหาลงอยู่ในวงแคบ ๆ จึงดีบักได้เร็ว (ข) สามารถใช้ส่วนที่เขียนรายละเอียดแล้วเป็นตัวช่วยในการทดสอบส่วนใหม่ ซึ่งจะทำให้ไม่ต้องเสียเวลาย้อนกลับมาทดสอบทั้งโปรแกรมซ้ำอีก และ (ค) เป็นการออกแบบโครงสร้างของโปรแกรมในลักษณะเป็นทั้ง Top Down และ Bottom Up ในเวลาเดียวกัน ซึ่งทำให้เกิดการแยกโปรแกรมออกเป็นโมดูล (Module) ที่ชัดเจน และเป็นผลให้มองเห็นโครงสร้างของโปรแกรมได้ดี

จากข้อดีต่าง ๆ ข้างบนนี้จะเห็นว่า ทั้งหมดที่กล่าวมานั้นต่างมีผลทำให้พัฒนาได้เร็วขึ้นทั้งนั้น เรื่องนี้เราพิสูจน์ได้ไม่ยาก ในตัวอย่างที่ยกขึ้นมาให้ดูนั้นก็จะเห็นได้ ผมใช้เวลาตอนหัวค่ำหลังทานอาหารเย็นเสร็จแล้ว ทำไปพร้อม ๆ กับการเขียนบทความนี้ทำอยู่ 4 หัวค่ำ ๆ ละ 1-2 ชม. ก็เสร็จเรียบร้อย แต่ที่ลึกลงไปกว่านี้ก็คือ การเขียนโปรแกรมด้วยวิธีการนี้จะลดความกดดันได้เป็นอย่างมาก **เหตุที่เราลดความกดดันได้ก็เพราะเราลดเป้าหมายลงเป็นการดูและทำให้เสร็จทีละโพรซีเดอร์** แต่ละโพรซีเดอร์มีขนาดและหน้าที่ที่แคบกว่าตัวโปรแกรมทั้งโปรแกรม มันลดขนาดลง 5-6 เท่าและลดความซับซ้อนลงเป็นร้อยเท่า เราสามารถประมาณเวลาได้ค่อนข้างชัดเจน และหลังจากทำเสร็จไปบางส่วนแล้วเราก็ยังได้ใช้มันเพื่อการทดสอบโมดูลต่อ ๆ สิ่งเหล่านี้ทำให้เกิดความเชื่อมั่นและลดความกังวลใจได้มหาศาล

ผมคิดว่า การเขียนโปรแกรมโดยไม่มีแรงกดดันทางจิตใจคือหัวใจของเทคนิคนี้ มันทำให้ทุกสิ่งทุกอย่างราบรื่นขึ้น เราไม่เคยจะให้ความสนใจกับ

```
      WRITE(EXTRACTEDFILE,MASTERBUFF);
      END;
    END;
  CLOSE(MASTERFILE);
  CLOSE(EXTRACTEDFILE);
  IF EXTRACTEDRECNO=200 THEN
    BEGIN
    GOTOXY(3,8);WRITE(' NO RECORD CAN BE REMOVED, PLEASE EXPAND FILE');
    GOTO E1;
    END;
  GOTOXY(3,8);WRITE(' DOING RELOAD, PLEASE WAIT  ');
  ASSIGN(EXTRACTEDFILE,'TP1D03.DAT');
  RESET(EXTRACTEDFILE);
  ASSIGN(MASTERFILE,'TP1D01.DAT');
  REWRITE(MASTERFILE);
  FOR MASTERRECNO:=1 TO EXTRACTEDRECNO DO
    BEGIN
    SEEK(EXTRACTEDFILE,MASTERRECNO);
    READ(EXTRACTEDFILE,MASTERBUFF);
    SEEK(MASTERFILE,MASTERRECNO);
    WRITE(MASTERFILE,MASTERBUFF);
    END;
  MASTERBUFF.STATUS:=0;
  MASTERBUFF.TRUSTNAME:='                                        ';
  MASTERBUFF.PHONENUMBER:='          ';
  MASTERBUFF.AMOUNT:=0.0;
  MASTERBUFF.RATE:=0.0;
  MASTERBUFF.STARTDAY:=0;
  MASTERBUFF.STARTMONTH:=0;
  MASTERBUFF.STARTYEAR:=0;
  MASTERBUFF.ENDDAY:=0;
  MASTERBUFF.ENDMONTH:=0;
  MASTERBUFF.ENDYEAR:=0;
  MASTERBUFF.PNNUMBER:='          ';
  MASTERBUFF.REFNO:='          ';
  MASTERBUFF.REMARK:='                                        ';
  FOR MASTERRECNO:=EXTRACTEDRECNO+1 TO 200 DO
    BEGIN
    SEEK(MASTERFILE,MASTERRECNO);
    WRITE(MASTERFILE,MASTERBUFF);
    END;
  CLOSE(MASTERFILE);
  CLOSE(EXTRACTEDFILE);
  GOTOXY(3,10);WRITE(' COMPLETED  ');
E1:
  DELAY(1000);
END;

BEGIN

MAINLOOP:
  NEWSCREEN;
  GOTOXY(3,3);WRITE(' TP1S07:MAIN PROGRAM  ');
  GOTOXY(10,5);WRITE(' 1.  UPDATE MASTERFILE  ');
  GOTOXY(10,7);WRITE(' 2.  SCAN   MASTERFILE  ');
  GOTOXY(10,9);WRITE(' 3.  MAKE   REPORT      ');
  GOTOXY(10,11);WRITE(' 4.  MAKE   BACK-UP     ');
  GOTOXY(10,13);WRITE(' 5.  EXTRACT/RELOAD MASTER ');
  GOTOXY(10,15);WRITE(' 9.  END JOB            ');
  REPEAT
    GOTOXY(3,23);WRITE('                        ');
    GOTOXY(3,24);WRITE('                        ');
    GOTOXY(3,23);WRITE(' YOUR SELECTION-->    ');
    GOTOXY(3,24);WRITE('       <1-5 OR 9>     ');
    READ(KBD,CH);
    UNTIL (CH='1') OR (CH='2') OR (CH='3') OR
          (CH='4') OR (CH='5') OR (CH='9');
  IF CH='1' THEN UPDATEMASTER;
  IF CH='2' THEN SCANMASTER;
  IF CH='3' THEN MAKEREPORT;
  IF CH='4' THEN MAKEBACKUP;
  IF CH='5' THEN EXTRACTRELOAD;
  IF CH='9' THEN GOTO EOJ;
  GOTO MAINLOOP;

EOJ:
  GOTOXY(50,24);WRITE('END OF JOB');
  END.
```

เรื่องนี้มาก่อน ทั้งนี้โดยไปยอมรับกันว่า การเขียนโปรแกรมนั้นเป็นงานที่ต้องทนทุกข์ทรมาน เราไปสร้างความอดทน เราไปสร้างขวัญและกำลังใจกัน ซึ่งก็พอจะช่วยกันไปได้บ้าง แต่การเขียนโปรแกรมโดยไม่มีแรงกดดันนี้ช่วยเราได้มากกว่า มันเป็นการแก้ปัญหาที่ตรงจุดมากที่สุด

สำหรับข้อเสียของเทคนิคนี้มีอยู่นิดเดียว คือ เราต้องเสียเวลาเขียนคำสั่งใส่ไว้ในโพรซีเดอร์ที่ยังไม่มีรายละเอียด เป็นจำนวนโพรซีเดอร์ละ 3-4 บรรทัด คำสั่งเหล่านี้ในที่สุดก็ต้องถูกลบทิ้งไป มันจึงเป็นการสูญเสียอย่างหนึ่ง แต่ท่านผู้อ่านก็คงจะมองเห็นได้ว่าแรงงานที่ต้องเสียไปกับการใส่คำสั่งเหล่านี้มันเป็นแค่เศษของสตางค์ ในการเขียนโปรแกรมโดยวิธีอื่น ๆ นั้น เราเขียนแล้วลบทิ้งไปมากกว่านี้มาก เรื่องนี้จึงเป็นการลงทุนที่เกินคุ้ม

บทสรุป

เทคนิคการเขียนโปรแกรมแบบ "ค่อยเป็นค่อยไป" ดังที่ได้กล่าวมานี้เกิดขึ้นได้ก็เพราะภาษาปาสคาลมีลักษณะที่เหมาะสม คือ เป็นภาษาที่ใช้แยกโปรแกรมเป็นโพรซีเดอร์ได้ดีมาก แต่อย่างไรก็ตาม หลักการสำคัญของเรื่องนี้ก็คือ "เมื่อเราแยกมาเพ่งมองทีละโพรซีเดอร์นั้นมันช่วยลดแรงกดดันทางจิตใจ เราสามารถควบคุมเวลาในการทำงานได้ เรามีความเชื่อมั่นว่าเราสามารถทำให้เสร็จได้ และหลังจากนั้นแล้วเราก็หมดกังวล เพราะเราสามารถใช้ส่วนที่เขียนเสร็จแล้วมาช่วยเช็คงานส่วนใหม่" เทคนิคนี้จึงไม่จำเป็นต้องจำกัดอยู่ แต่ในภาษาปาสคาล ท่านผู้อ่านที่ชำนาญในภาษาอื่นคงจะนำไปปรับใช้ได้ ผมเชื่อว่าบทความนี้คงจะมีประโยชน์กับพวกเราอยู่บ้าง

หมายเหตุ โปรแกรม TP1Sϕ3.PAS, TP1Sϕ4.PAS, TP1Sϕ5.PAS และ TP1ϕ6.PAS จะอยู่ในบริการซอฟต์เมล์ เราไม่สามารถนำลงมาพิมพ์ได้หมด เพราะจะทำให้บทความยาวเกินไป แต่โพรซีเดอร์ต่าง ๆ ที่กล่าวถึงก็มีอยู่ในรูปที่ 4 ซึ่งจะดูจากรูปนี้เลยก็ได้